DM 1 – David LOPES DIAS

Exercice 1 : Définir une fonction qui renvoie la position du plus grand élément d'un tableau.

Le but de cet exercice est d'écrire une algorithme général qui a pout but de retourner la position de l'élément le plus élevé d'un tableau. Peu importe si ce tableau est, ou pas, trié de n'importe quelle manière que ce soit.
Dans ce cas, on entre :

- Un tableau qui contient des nombres, trié ou non

Et en sortie on obtient :

- La position du nombre le plus élevé

Voici la fonction :

```
def sor(tab):
    # variable pour stocker la valeur max contenue dans le tableau
    max = 0
    # variable pour stocker l'indice du tableau auquel l'élément max se situe
    compteur = 0
    # premier tour de boucle pour trouver quelle est la valeur max
    for i in range(len(tab)):
        if (tab[i] >= max):
            max = tab[i]
    # deuxième tour de boucle pour trouver la position de cette valeur max
    for j in range(len(tab)):
        if (tab[j] == max):
            compteur = j
    # On retourne la valeur de l'indice
    return compteur
```

Exercice 2 : Faire une trace d'exécution de cette fonction sur le tableau [5,3,8,5,9]

Ligne	I	J	Tableau
8	5	/	[5,3,8,5,9]
9	5	/	[5,3,8,5,9]
8	3	/	[5,3,8,5,9]
9	3	/	[5,3,8,5,9]
8	8	/	[5,3,8,5,9]
9	8	/	[5,3,8,5,9]
8	5	/	[5,3,8,5,9]
9	5	/	[5,3,8,5,9]
8	9	/	[5,3,8,5,9]
9	9	/	[5,3,8,5,9]
11		0	[5,3,8,5,9]
12		0	[5,3,8,5,9]
11		1	[5,3,8,5,9]
12		1	[5,3,8,5,9]

11		2	[5,3,8,5,9]
12		2	[5,3,8,5,9]
11		3	[5,3,8,5,9]
12		3	[5,3,8,5,9]
11		4	[5,3,8,5,9]
12		4	[5,3,8,5,9]
14	RETURN	RETURN	[5,3,8,5,9]

La fonction retourne la position de 9, c'est-à-dire 4.

Donc, la valeur la plus élevée, dans ce tableau, se situe à la position 4.

Exercice 3 : Définir une fonction qui donne la plus grande valeur du tableau.

Le but de cet exercice est de déduire un algorithme qui renvoie, non pas la position, mais la valeur la plus grande pour un tableau donné.

En entrée :

- Un tableau, qui contient des nombres, qui n'est pas forcément trié

En sortie :

- La valeur du nombre le plus élevé. Autrement dit, le nombre le plus élevé.

Voici la fonction :

```
def val_max(tab):
    # variable pour stocker la valeur max contenue dans le tableau
    max = 0
    # boucle pour trouver la valeur max
    for i in range(len(tab)):
        if (tab[i] >= max):
            max = tab[i]
    # On retourne la valeur max
    return max
```

Exercice 4 : Faire cette même fonction, mais dans un tableau qui peut contenir deux fois le même nombre, qui est le plus grand élément.

Entrées :

- Un tableau qui n'est pas trié, et qui contient deux nombres égaux qui ont la valeur la plus élevées du tableau

Sortie :

- La position de ces deux éléments dans ce tableau

```
def ind_max(tab):
    # variable pour stocker la valeur max contenue dans le tableau
    max = 0
    # variable pour stocker l'indice du tableau auquel l'élément max se situe
    compteur = []
    # premier tour de boucle pour trouver quelle est la valeur max
    for i in range(len(tab)):
        if (tab[i] >= max):
            max = tab[i]
    # deuxième tour de boucle pour trouver à quelle position sont ces
valeurs
    for j in range(len(tab)):
        if (tab[j] == max):
            x = j
            compteur.append(x)
    # On retourne la position
    return compteur
```

Exercice 5 : Faire une trace d'exécution de la précédente fonction pour le tableau [5,3,8,5,8]

Ligne	I	J	Tableau
8	5	/	[5,3,8,5,8]
9	5	/	[5,3,8,5,8]
8	3	/	[5,3,8,5,8]
9	3	/	[5,3,8,5,8]
8	8	/	[5,3,8,5,8]
9	8	/	[5,3,8,5,8]
8	5	/	[5,3,8,5,8]
9	5	/	[5,3,8,5,8]
8	8	/	[5,3,8,5,8]
9	8	/	[5,3,8,5,8]
12		0	[5,3,8,5,8]
13		0	[5,3,8,5,8]
14		0	[5,3,8,5,8]
12		0	[5,3,8,5,8]
13		0	[5,3,8,5,8]
14		0	[5,3,8,5,8]
12		1	[5,3,8,5,8]
13		1	[5,3,8,5,8]
14		1	[5,3,8,5,8]
12		0	[5,3,8,5,8]
13		0	[5,3,8,5,8]
14		0	[5,3,8,5,8]
12		1	[5,3,8,5,8]
13		1	[5,3,8,5,8]
14		1	[5,3,8,5,8]
15	RETURN	RETURN	[5,3,8,5,8]

Donc, on peut en déduire que dans le tableau [5,3,8,5,8], 8 est la valeur la plus grande et est présente 2 fois, qui sont, respectivement aux positions 2 et 4 .

Exercice 6 : On suppose que l'on a un tableau trié, réécrire les fonctions précédentes en les simplifiant.

Entrées :

- Un tableau trié par ordre croissant contenant des nombres

Sortie :

- La position du nombre avec la plus haute valeur

1 .
```
def val_max(tab):
    # le tableau étant déjà trié, le dernier élément a forcément la valeur maximale
    return tab[len(tab)-1]
```

Si on considère le tableau : cacao = [5,3,1,5,8]

Dans ce cas, la fonction renverra 4.

2.

Entrées :

- Un tableau trié par ordre croissant contenant des nombres

Sortie :

- La position du nombre avec la plus haute valeur

```
def sor(tab):
    # On retourne la valeur du dernier indice qui correspond à la longueur du tableau - 1
    return (len(tab)-1)
```

Si on considère le précédent tableau cacao, cette fonction renvoit 4.

3.
Entrée :

- Un tableau, contenant des nombres, dont le plus élevé est présent deux fois

Sortie :

- La position à laquelle se situe ces deux nombres les plus élevés.

```
def ind_max(tab):
    # variable pour stocker la valeur max contenue dans le tableau
    max = 0
    # variable pour stocker les indices auxquels se situent les éléments max
    compteur = []
    # premier tour de boucle pour trouver quelle est la valeur max
    for i in range(len(tab)):
        if (tab[i] >= max):
            max = tab[i]
    j = 0
    # On cherche le premier indice auquel le maximum est présent
    while (tab[j] != max):
        j += 1

    # On ajoute les indices des éléments qui ont la valeur max
    for z in range (j, len(tab)):
        compteur.append(z)

    return compteur
```

Exercice 7 : Le but de cet exercice est de définir une fonction qui inverse un tableau donné.

Entrée :

- Un tableau, trié ou non, contenant des nombres

Sortie :

- Ce même tableau mais inversé.

Voici la fonction :

```
def inv(tab):
    # On crée un tableau vide
    tmp = []
    taille = len(tab)
    # On remplit le tableau tmp avec les mêmes éléments que le tableau passé
en entrée
    for j in range (taille):
        tmp.append(tab[j])

    # On fait correspondre l'élément du tableau temporaire avec son inverse
dans le tableau passé en paramètre
    for i in range (taille):
        tmp[i] = tab[taille - i - 1]

    return tmp
```

Si on considère le tableau « cacao », alors cette fonction nous renvois son inverse, c'est-à-dire :[8,5,1,3,5]

Exercice 8 : L'exécuter sur le tableau [5,3,8,5,9]

Ligne	J	I	Tableau
6	5	/	[5,3,8,5,9]
7	5	/	[5,3,8,5,9]
6	3	/	[5,3,8,5,9]
7	3	/	[5,3,8,5,9]
6	8	/	[5,3,8,5,9]
7	8	/	[5,3,8,5,9]
6	5	/	[5,3,8,5,9]
7	5	/	[5,3,8,5,9]
6	9	/	[5,3,8,5,9]
7	9	/	[5,3,8,5,9]
9		9	[5,3,8,5,9]
10		9	[5,3,8,5,9]
9		5	[5,3,8,5,9]
10		5	[5,3,8,5,9]
9		8	[5,3,8,5,9]
10		8	[5,3,8,5,9]
9		3	[5,3,8,5,9]
10		3	[5,3,8,5,9]
9		5	[5,3,8,5,9]
10		5	[5,3,8,5,9]
11	Return	Return	[9,5,8,3,5]

Donc, cette fonction retourne l'inverse du tableau en entrée. Ce qui se traduit par l'insertion de la variable « -i ».